

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-167517

(43) 公開日 平成11年(1999) 6月22日

(51) Int.Cl.⁶

G 0 6 F 12/00

識別記号

5 7 0

F I

G 0 6 F 12/00

5 7 0 B

審査請求 未請求 請求項の数 3 O L (全 13 頁)

(21) 出願番号

特願平9-334239

(22) 出願日

平成9年(1997)12月4日

(71) 出願人 000004075

ヤマハ株式会社

静岡県浜松市中沢町10番1号

(72) 発明者 藤田 佳生

静岡県浜松市中沢町10番1号 ヤマハ株式会社内

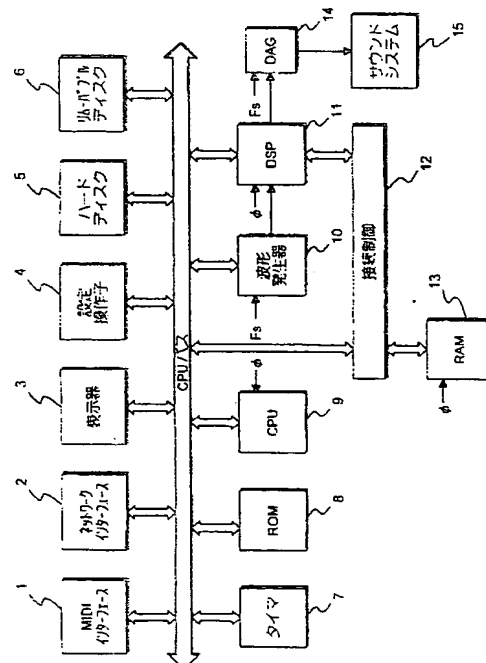
(74) 代理人 弁理士 川▲崎▼ 研二 (外1名)

(54) 【発明の名称】 信号処理装置

(57) 【要約】

【課題】 CPUの処理速度を低下することなく、CPUとDSPでメモリを共有する。

【解決手段】 CPU9はバッファ121を介してRAM13にアクセスできるようになっており、CPU9からのアドレスとDSP11からのアドレスがセクタ122に供給される。セクタ122はCPU9によって生成される制御信号CS2によって制御される。これにより、CPU9がRAM13にアクセスする際に、DSP11に優先される。一方、DSP11は、CPU9がRAM13にアクセスしていないことを制御信号CS2によって検知し、空きタイムスロットを利用してRAM13にアクセスする。この場合、DSP11は内部に設けられたFIFOを介してRAM13にアクセスする。これにより、DSP11のデータ書込読出タイミングが調整される。



【特許請求の範囲】

【請求項1】 装置全体の制御を行う制御プロセッサと、予め定められた演算処理を実行する信号処理プロセッサと、前記制御プロセッサと前記信号処理プロセッサとでアクセス可能な共有メモリとを有する信号処理装置において、

前記制御プロセッサから前記共有メモリへのアクセスを、前記信号処理プロセッサからの前記共有メモリへのアクセスに優先させるように制御するアクセス制御部を備え、

前記信号処理プロセッサは、
処理動作を規定するプログラムを格納するプログラムメモリと、

前記プログラムメモリから前記プログラムを読み出す読出部と、

前記プログラムに従って入力データに演算処理を施して出力データを生成するとともに、前記プログラムで指定されるタイミングにおいて前記出力データの生成過程で生ずる中間データを入出力する演算部と、

前記演算部からの前記中間データを前記共有メモリに出力する際に当該中間データをバッファリングするとともに、前記共有メモリからの前記中間データを前記演算部に出力する際に当該中間データをバッファリングするバッファ部と、

前記制御プロセッサが前記共有メモリにアクセスしていない期間を検知し、当該期間において前記バッファ部からの前記中間データを前記共有メモリに入出力するように前記バッファ部を制御するとともに、前記プログラムで指定されるタイミングにおいて前記バッファ部からの前記中間データを前記演算部に入出力するように前記バッファ部を制御する制御部とを有し、

前記演算部と前記バッファ部との間の前記中間データの入出力は、前記プログラムで指定されるタイミングで行うことを特徴とする信号処理装置。

【請求項2】 前記制御部は、前記中間データを前記共有メモリに書き込むための書込アドレスと、前記共有メモリから前記中間データを読み出すための読出アドレスとを発生し、前記制御プロセッサが前記共有メモリにアクセスしていない期間において、前記書込アドレスまたは前記読出アドレスを前記共有メモリに出力することにより、前記バッファ部から前記中間データを前記共有メモリに入出力することを特徴とする請求項1に記載の信号処理装置。

【請求項3】 記入力データと前記出力データはサンプルクロックに同期して前記演算部から入出力され、前記演算部は、前記サンプルクロック周波数の整数倍の周波数であるシステムクロックに基づいて、1サンプルクロック周期毎に複数の演算処理を実行し、

前記制御部は、1サンプルクロック期間内に必要とされる前記共有メモリへのアクセス回数が満たされないこと

を事前に検知し、前記制御プロセッサに対して前記共有メモリへのアクセスを停止するよう指令することを特徴とする請求項1または2に記載の信号処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、CPUとDSPでRAMを兼用するのに好適な信号処理装置に関する。

【0002】

【従来の技術】電子楽器では、波形発生器から出力される波形信号にデジタル処理を施して、所望の出力波形信号を得て、これをサウンドシステムに供給して発音を行うことが行われる。この場合、電子楽器の各構成部分を制御するCPU（中央演算処理装置）とは別に、波形信号にデジタル処理を施すDSP（デジタルシグナルプロセッサ）を用いて、リアルタイムに波形処理が施されることが多い。

【0003】

【発明が解決しようとする課題】ところで、CPUは、その作業領域としてのRAMと接続され、必要に応じてRAMにアクセスし、データの読出書込を行う。一方、DSPは、波形処理の演算アルゴリズムで必要とされるデータ遅延のため、演算途中のデータをRAMに一旦書き込んで、これを読み出して遅延されたデータを得ることがある。この場合、CPUとDSPに個別にRAMを設けると、2つのRAMが必要となるので、回路が大規模となり、さらに配線も複雑になるという欠点がある。

【0004】そこで、この点を解消するため、2つのプロセッサでRAMを兼用する技術が開発されている。この技術は、RAMへのアクセスに関するタイムスロットを時分割で行うものであって、予めタイムスロットをCPUとDSPに交互に割り振っておくものである。また、固定のタイムスロットで時分割を行う代わりに、DSPとCPUのRAMアクセスが重なったときにCPUにWAITをかけ、DSPによるRAMアクセスを優先させて行い、CPUはDSPによるアクセスが行われていない期間を利用してRAMアクセスを行うようにする方法も提案されている。しかしながら、CPUに連続してRAMをアクセスする必要性が生じても、DSPに割り振られたタイムスロットでは、CPUはRAMにアクセスすることができないので、それだけCPUは待たされることになる。この結果、CPUのRAMに対するアクセス速度が低下し、これに伴ってCPUの動作速度が低下する。したがって、CPUは本来の性能を十分発揮することができないといった問題があった。また、CPUによるRAMアクセスをDSPに優先させることも考えられるが、DSPはシステムクロックに同期してマイクロプログラムを実行することにより動作しており、CPUのようにWAITをかけることができないので、従来のDSPでそれを実現することは不可能であった。

【0005】本発明は、上述した事情に鑑みてなされた

ものであり、動作速度を低下させることなく、2つのプロセッサで1つのメモリを共有することを目的とする。

【0006】

【課題を解決するための手段】上記課題を解決するため、請求項1に記載の発明にあっては、装置全体の制御を行う制御プロセッサと、予め定められた演算処理を実行する信号処理プロセッサと、前記制御プロセッサと前記信号処理プロセッサとでアクセス可能な共有メモリとを有する信号処理装置において、前記制御プロセッサから前記共有メモリへのアクセスを、前記信号処理プロセッサからの前記共有メモリへのアクセスに優先させるように制御するアクセス制御部を備え、前記信号処理プロセッサは、処理動作を規定するプログラムを格納するプログラムメモリと、前記プログラムメモリから前記プログラムを読み出す読出部と、前記プログラムに従って入力データに演算処理を施して出力データを生成するとともに、前記プログラムで指定されるタイミングにおいて前記出力データの生成過程で生ずる中間データを入出力する演算部と、前記演算部からの前記中間データを前記共有メモリに出力する際に当該中間データをバッファリングするとともに、前記共有メモリからの前記中間データを前記演算部に出力する際に当該中間データをバッファリングするバッファ部と、前記制御プロセッサが前記共有メモリにアクセスしていない期間を検知し、当該期間において前記バッファ部からの前記中間データを前記共有メモリに入出力するように前記バッファ部を制御するとともに、前記プログラムで指定されるタイミングにおいて前記バッファ部からの前記中間データを前記演算部に入出力するように前記バッファ部を制御する制御部とを有し、前記演算部と前記バッファ部との間の前記中間データの入出力は、前記プログラムで指定されるタイミングで行うことを特徴とする。

【0007】また、請求項2に記載の発明にあっては、前記制御部は、前記中間データを前記共有メモリに書き込むための書込アドレスと、前記共有メモリから前記中間データを読み出すための読出アドレスとを発生し、前記制御プロセッサが前記共有メモリにアクセスしていない期間において、前記書込アドレスまたは前記読出アドレスを前記共有メモリに出力することにより、前記バッファ部から前記中間データを前記共有メモリに入出力することを特徴とする。

【0008】また、請求項3に記載の発明にあっては、前記入力データと前記出力データはサンプルクロックに同期して前記演算部から入出力され、前記演算部は、前記サンプルクロック周波数の整数倍の周波数であるシステムクロックに基づいて、1サンプルクロック周期毎に複数の演算処理を実行し、前記制御部は、1サンプルクロック期間内に必要とされる前記共有メモリへのアクセス回数が満たされないことを事前に検知し、前記制御プロセッサに対して前記共有メモリへのアクセスを停止す

るよう指令することとを特徴とする。

【0009】

【発明の実施の形態】A. 実施形態の構成

本実施形態においては、本発明に係わる信号処理装置の一例として、電子楽器を取りあげ説明する。

1. 実施形態の全体構成

図1は、本発明の一実施形態である電子楽器の全体構成を示すブロック図である。図において、1はMIDIインターフェースであって、MIDIデータの入出力を行う。2はネットワークインターフェースであって、通信回線を介して発音データや制御信号の送受信を行う。3は表示器であって、音量やMIDIデータによって指定される音色等がそこに表示される。4は設定操作子であって、演奏者がこれを実行すると、音色や音量あるいは各種のエフェクトが指定できるようになっている。5はハードディスクであって、MIDIデータ等が格納される。6は、リムーバブルディスクであって、フロッピーディスク等の可搬型の記録媒体が該当する。このリムーバブルディスクには、MIDIデータ等が図示せぬ読出書込装置を介して書き込まれ、また、読み出される。7はタイマであって、時間の計測を行う。8はROMであって、そこには装置全体を制御する制御プログラム等が格納されている。

【0010】次に、9はCPUであって、CPUバス16を介して装置の各部分と接続され、制御プログラムに従って装置全体を制御する。また、CPU9は、MIDIインターフェース1からの演奏入力に応じて、波形発生器10やDSP11で生成する楽音の制御を行う。例えば、ノートオンが入力した場合、このノートオンに対し、波形発生器10の発音チャンネルを割当て、割り当てたチャンネルにおいて当該ノートオンに対応した楽音の発生を指示する。

【0011】次に、波形発生器10は、複数の発音チャンネルを有しており、それぞれ時分割動作を行う低周波発振器OSC、デジタルフィルタDCF、エンベロープ発生器EG、およびチャンネル累積器ACC等で構成される。各発音チャンネルは、各々独立しており、複数の楽音データを発生できるようになっている。発生された楽音データは混合され、波形データとして出力される。ここで波形データのサンプル周波数は、44.1KHzであり、波形発生器10には、その周波数が44.1KHzのサンプルクロックFsが供給される。また、CPU9は、サンプルクロックFsに同期したシステムクロックφで動作しており、その周波数は5.6448MHz(=128*Fs)に設定されている。したがって、CPU9は波形データの1サンプル期間に128ステップの処理を行うことができる。なお、システムクロックφは図示せぬ水晶発振回路によって生成され、これを128分周してサンプルクロックFsが生成される。

【0012】また、DSP11は、波形発生器10から

の波形データに加減乗除等の演算処理を施す。これにより、リバーブ、ディストーションまたはコーラスといったエフェクト効果を得ることができる。ここで、DSP 11はシステムクロック ϕ によって動作する。上述したようにシステムクロック ϕ はサンプルクロックFsに同期しており、 $\phi = 1/28Fs$ の関係があるから、DSP 11は、波形データに同期して1サンプル当たり128ステップの演算処理を行うことが可能である。そして、DSP 11で波形処理を施された波形データはDAコンバータ(DAC)14によってアナログ信号に変換され、これがサウンドシステム15に供給されると、発音されるようになっている。なお、本実施例では、簡単化のために(マイクロプログラムのステップ数) = (1サンプリング周期内の遅延メモリアクセス数)として説明を行うが、必ずしもそうである必要はない。DSPの遅延メモリに対するアクセス頻度を、マイクロプログラムの3ステップに1回とか、4ステップに1回にしても良い。

【0013】次に、12は接続制御部であって、CPUバス16およびDSP 11とRAM 13に接続される。この接続制御部12は、CPU9からRAM 13へのアクセスをDSP 11からRAM 13へのアクセスに優先させるように制御する。また、RAM 13はCPU9の作業領域として使用されるとともに、DSP 11からのデータを遅延させる遅延メモリとして動作する。

【0014】2. 主要部の構成

図2は、本実施形態に係わる電子楽器の主要部の構成を示すブロック図である。図において、CPUバス16は、コントロールバスb1、読出書込バスb2、アドレスバスb3、およびデータバスb4から構成される。そして、CPU9は、読出動作または書込動作を指示する判定信号RWを読出書込バスb2に出力し、アドレス信号ADをアドレスバスb3に出力し、読出動作ではデータバスb4上のデータDを取り込むと共に、書込動作では書き込もうとするデータDをデータバスb4に出力する。判定信号RWはハイレベルで読出を指示し、ローレベルで書込を指示する。また、アドレス信号ADは、RAM 13、ROM 8、DSP 11、MIDIインターフェース1といったアクセス先の別と、該アクセス先の内部における個別のアドレス指定を行う。

【0015】また、アドレスデコーダ16はアドレス信号ADをデコードして、アクセス先を指示する制御信号CS1~CS3を生成する。各制御信号CS1~CS3はハイレベルでアクティブを指示する。例えば、CPU9がRAM 13にアクセスする場合には、制御信号CS2をハイレベルにして、その他の構成部分に送出する制御信号CS1、CS3をローレベルにする。これによって、アドレスバスb3上のアドレス信号ADがRAM 13に対するアドレスであることが判る。また、図1のMIDIインターフェース1等、その他のブロックについ

ても同様であり、各ブロックにはそれぞれアドレスデコーダが備えられており、それぞれ対応するアドレスによりアクセスされる。

【0016】次に、DSP 11は、CPU9の制御の下、波形データの演算処理を行う。演算処理は、マイクロプログラムと制御データに基づいて行われる。DSP 11には、CPU9からCPUバス16を介して、マイクロプログラムと制御データが転送される。また、DSP 11はRAM 13のデータ入出力端子と接続されており、RAM 13に演算途中のデータDを一旦書き込み、所定時間が経過した後これを読み出すようになっている。これにより、DSP 11はRAM 13を遅延メモリとして使用する。なお、DSP 11の詳細な構成については後述する。

【0017】次に、接続制御部12はバッファ121とセクタ122によって構成される。バッファ121は、スリーステート形式で構成されており、端子dirのレベルと端子ENのレベルに基づいて、端子Aから端子BにデータDを伝送するモード、端子Bから端子AにデータDを伝送するモード、あるいは端子A、Bをともにハイインピーダンス状態とするモードを各々選択できるようになっている。

【0018】まず、端子A、Bをともにハイインピーダンス状態とするか否かは、端子ENのレベルに基づいて制御される。端子ENのレベルがハイレベルであれば、データ伝送が行われ、一方、端子ENのレベルがローレベルであれば、端子A、Bをともにハイインピーダンス状態となる。このため、端子ENのレベルがローレベル期間中は、データ伝送が行われない。この端子ENには、図に示すように制御信号CS2が供給されるので、CPU9がRAM 13にアクセスする期間にデータ伝送が行われ、他の期間ではRAM 13とデータバスb4との間が非接続状態となる。なお、DSP 11からRAM 13へのアクセスは、当該他の期間を利用して行われる。

【0019】また、セクタ122は2つの信号のうちいずれか一方を選択して出力する選択回路を2系統備えたものである。その端子Sがハイレベルの場合、セクタ122は入力端子B1、B2に供給される信号を出力端子Y1、Y2に出力する。ここで、端子Sには、CPU9がRAM 13にアクセスする場合にのみハイレベルとなる制御信号CS2が供給されるから、CPU9がRAM 13にアクセスしようとする、CPU9からの判定信号RWが出力端子Y1を介してRAM 13に出力されるとともに、CPU9からのアドレス信号ADが出力端子Y2を介してRAM 13に供給される。この場合には、上述したようにバッファ121は、データ伝送を行う。したがって、CPU9は、必要とされる時に制御信号CS2をハイレベルにしてRAM 13にアクセスすることができる。このため、CPU9はDSP 11に優先してRAM 13を使用

することができる。一方、端子Sがローレベルの場合、セクタ122は入力端子A1、A2に供給される信号を出力端子Y1、Y2に出力すると共にバッファ121が非接続状態となり、DSP11はRAM13に対するアクセスが可能になる。

【0020】3. DSPの構成

次に、DSPの詳細な構成を説明する。図3はDSPの詳細な構成を示すブロック図である。図に示すように、DSP11は、CPUバス16と接続されるCPUインターフェース112を備えており、これによって、CPU9から送信されるマイクロプログラムや、制御データ等の受信が行われる。マイクロプログラムは、波形演算部115で行う各種の演算処理やRAM13との間でデータの読出書込処理を指示する。また、制御データは、波形演算部115で行う演算処理の定数やRAM13への書込読出の基準となる基準書込アドレスWA_r、基準読出アドレスRA_rを指定する。なお、実際にRAM13へアクセスする際に用いる読出アドレスRAと書込アドレスWAは、基準書込アドレスWA_r、基準読出アドレスRA_rに基づいて後述するアドレス発生器118で生成される。

【0021】受信したマイクロプログラムはマイクロプログラムメモリ113に格納され、一方、受信した制御データは制御レジスタ111に格納される。制御レジスタ111は、128個のレジスタから構成されており、各レジスタに制御データが各々格納される。そして、システムクロックφに同期して各レジスタから制御データが順次読み出されるようになっている。

【0022】次に、読出回路114は、7ビットのリングカウンタ等で構成される。このリングカウンタでシステムクロックφをカウントすると、そのカウント値がアドレスとして出力される。この場合、アドレスは0～127を巡回する。このアドレスがマイクロプログラムメモリ113に供給されると、マイクロプログラムが1ステップ毎順次読み出される。

【0023】ここで、マイクロプログラムの一例を図4に示す。図に示すようにマイクロプログラムは128ステップで構成されており、システムクロックに同期した各ステップ毎の各構成部分の動作を規定している。サンプルクロックφの周波数は、上述したようにサンプルクロックFsの128倍に設定されているから、第0ステップ～第127ステップのプログラムは1サンプルクロックFs内で読み出される。また、読出回路114で生成されるアドレスは0～127を巡回するから、第127ステップの読出が終了すると、その次には第0ステップに戻ってマイクロプログラムが読み出される。この図において第9ステップに示すINは、波形入力部116でラッチされた波形データを波形演算部115に取り込むことを指示しており、一方、第124ステップに示すOUTは、波形演算部115から波形データを出力する

ことを指示している。すなわち、この例によれば、波形演算部115には1サンプルクロックFs毎に、波形データの入出力を行う指示がなされる。なお、DSP11は、上述したINおよびOUTのタイミングで波形データの入出力を行いながらRAM13のアクセスを行うことも可能である。また、音源10が複数系統の出力を有している場合には複数ステップで複数の波形データの入力を行い、DACおよびサウンドシステムがステレオの場合には2ステップで2つの波形データの出力を行う。

【0024】また、第6、第12、第127ステップのW1、W2、W17は書込命令であって、このタイミングでデータDを書き込むことを指示する。読出回路114は書込命令を判別して、書込期間でハイレベルとなり、他の期間でローレベルとなる書込信号Wを生成している。また、第1～第3、第7、第122ステップのR1～R4、R39は、読出命令であって、このタイミングでデータDを読み出すことを指示する。読出回路114は読出命令を判別して読出期間でハイレベルとなり、他の期間でローレベルとなる読出信号Rを生成している。

【0025】また、読出命令に同期して基準読出アドレスRA_rが制御レジスタ111から読み出され、また、書込命令に同期して基準書込アドレスWA_rが制御レジスタ111から読み出される。逆にいえば、マイクロプログラムの進行に応じて必要な制御データが読み出せるように、制御レジスタ111には各制御データが格納されている。

【0026】次に、波形演算部115では、読出回路114から出力される命令に基づいて、各種の演算処理が実行され、これによって、リバーブやディストーションといったエフェクト効果を得ることができる。ところで、リバーブは音を響かせるものあって、例えば、過去の波形レベルに係数を乗算したものを現在のレベルと加算することによって得られる。この場合には、遅延した波形データに係数を乗算し、これと現在の波形データを加算することが行われる。このように、波形演算部115で行われる演算処理には、過去の波形データが必要とされる場合がある。ここで、遅延時間が短時間であれば波形演算部115の内部レジスタが使用されるが、遅延時間が長時間の場合にはRAM13が使用される。なお、波形演算部115から出力されるMDは変調データであって、これによって、波形データに周波数変調ないし位相変調を施すことができる。

【0027】次に、アドレス発生器118について説明する。図5はアドレス発生器の回路図である。なお、この例では、RAM13の全記憶領域のうちスタートアドレスADsで始まる256k(18bit)ワードの記憶領域をDSP11に割り当てている。図において、カウンタ20は18ビットのリングカウンタである。そのクロック入力端子にはサンプルクロックFsが供給され、この

カウンタ20によって、サンプルクロックFsが0から256k-1までカウントされる。ここで、カウント値が256k-1の状態において、次のサンプルクロックFsが入力すると、カウント値が0に戻る。したがって、カウント値は0から256k-1までの値を巡回する。

【0028】次に、加算器21は、カウンタ20のカウント値と基準書込アドレスWA_rまたは基準読込アドレスWA_rと、変調データMDの加算を行う。この加算器21は18ビットのフルアダーで構成される。ところで、上述したようにカウンタ20のカウント値は18ビット、基準書込アドレスRA_rは7ビットである。また、変調データMDは例えば8ビットである。したがって、加算器21の加算結果は18ビットを越えることがあるが、19ビット以上は用意されていないので、上位ビットがマスクされることになる。

【0029】次に、加算器22は加算器21の出力とスタートアドレスAD_rと加算して、実際にRAM13にアクセスするための書込アドレスWA、読出アドレスRAを生成してこれを図3に示すFIFOA121のデータ入力端子に出力する。

【0030】例えば、1サンプルクロックFs当たり3回のデータ書込読出を行うものとし、これらのデータ書込に用いる基準書込アドレスWA_rを、WA_r1、WA_r2、WA_r3、データ読出に用いる基準読出アドレスRA_rを、RA_r1、RA_r2、RA_r3とする。いま、カウンタ20のカウント値が0で、また変調データMDを無視するものとする。この状態において加算器22から出力される書込アドレスWAは、WA1=WA_r1+AD_s、WA2=WA_r2+AD_s、WA3=WA_r3+AD_sとなる。ここで、カウンタ20にサンプルクロックFsが100個入力したとすると、書込アドレスWAは、WA1=WA_r1+AD_s+100、WA2=WA_r2+AD_s+100、WA3=WA_r3+AD_s+100となり、各書込アドレスWA1~WA3で指定される記憶領域にデータD1~D3が書き込まれる。

【0031】ここで、基準読出アドレスRAが、RA_r1=WA_r1-10、RA_r2=WA_r2-20、RA_r3=WA_r3-30であって、カウンタ20にサンプルクロックFsが100個入力した時点でデータDを読み出すものとするれば、その際の読出アドレスRAは、RA1=WA_r1+AD_s+90、RA2=WA_r2+AD_s+80、RA3=WA_r3+AD_s+70となる。このため、90番目サンプル、80番目サンプル、70番目サンプルに各々対応したデータDが読み出される。これによって、RAM13を遅延メモリとして使用し、過去の演算結果を利用して現在の演算を実行することができる。

【0032】この場合、例えば、WA_r1=0、WA_r2=0、WA_r3=0とすれば、WA1=AD_s、WA2=AD_s、WA3=AD_sとなり、各書込アドレスWA1~WA3で指定される記憶領域にデータD1~D3が書き込まれる。

2=80k、WA_r3=120kに設定すれば、それぞれサンプルクロックFsが80k、40k、136k個入力されて始めてRAM13上の書込位置が重複する。したがって、RA_r1~3の設定に応じて、それぞれ最大80k、40k、136kサンプル前の波形データを取り出して演算を行うことができる。

【0033】ところで、上述したようにRAM13へのアクセスは、CPU9からのものが、DSP11に優先する。したがって、DSP11のRAM13へのアクセスは、CPU9がアクセスしていないタイムスロットで実行する必要がある。しかし、CPU9のアクセスは、処理の必要に応じてランダムに発生するため、1サンプルクロックFs内の空きスロットのタイミングは、その都度変動する。一方、DSP11は、マイクロプログラムに従って必要なデータの書込読出を行う必要がある。このため、DSP11のアクセスタイミングがCPU9のアクセスタイミングと一致してしまうことがある。そこで、本実施形態にあっては、DSP11がCPU9によるアクセスが行われていない空きスロットでRAM13にアクセスできるように、FIFO（ファーストイン・ファーストアウトメモリ）を使用して、タイミングを調整している。

【0034】図3に示すFIFOA121、FIFOB122、およびFIFOC123は、入力順にデータを出力するファーストイン・ファーストアウトメモリであって、シフトレジスタによって構成される。FIFOA~Cは、ハイレベルでアクティブとなる入力イネーブル端子INと出力イネーブル端子OUTを備え、システムクロックφに同期して動作する。FIFOA121はアドレスデータと書込または読出の種別を指定する判定信号用のメモリ、FIFOB122は出力データ用のメモリ、また、FIFOC123は入力データ用のメモリである。

【0035】まず、FIFOA121の周辺回路を説明する。読出回路114からマイクロプログラムが読み出され、書込信号Wと読出信号Rが生成されると、これが判定信号生成部119に供給され、そこで判定信号RWが生成される。判定信号RWは、書込信号Wが生成されたときはローレベルとなり、読出信号Rが生成されたときにはハイレベルとなる。また、論理和回路120は書込信号Wと読出信号Rの論理和を算出し、アクセス信号を生成する。書込信号Wと読出信号Rは、上述したように書込期間と読出期間で各々ハイレベルとなるから、アクセス信号がハイレベルの期間は、波形演算部115がデータDの書込または読出を要求する期間である。FIFOA121の入力イネーブル端子INには、図に示すようにアクセス信号が供給される。したがって、FIFOA121には、判定信号RWと、書込アドレスWAおよび読出アドレスRAがマイクロプログラムに従って順次記憶されていく。また、書込信号Wがハイレベルにな

るとき、マイクロプログラムに従って波形演算部115からRAM13に書き込もうとするデータDが出力されてFIFOB122に記憶される。一方、読出信号Rがハイレベルのときは、2サンプリング周期前の同タイムスロットで指定された読出アドレスに応じてRAM13から読み出されたデータDがFIFOC123から出力されるので、波形演算部115はそのデータDをマイクロプログラムに従って内部に取り込み演算に使用する。該データDをRAM13からFIFOC123に取り込む部分については、図6ないし図7に関連して後で説明する。

【0036】次に、送り信号発生回路124は、CPU9からの制御信号CS2に基づいて、FIFOA121およびFIOB122のデータ出力と、FIFOC123のデータ入力を各々制御するアクセス制御信号AC1～AC3を各々生成する。具体的には、制御信号CS2がローレベルの期間（CPU9がRAM13にアクセスしない期間）を特定し、この期間においてRAM13にアクセスできるようにしている。

【0037】まず、制御信号CS2がローレベルになると、送り信号発生回路124はアクセス制御信号AC1をハイレベルにする。そして、アクセス制御信号AC1がFIFOA121の出力イネーブル端子OUTに供給されると、FIFOA121から判定信号RWと書込アドレスWAまたは読出アドレスRAが出力される。判定信号RWは接続制御部12を介してRAM13に供給されるとともに、送り信号発生回路124にフィードバックされる。この後、送り信号発生回路124は、判定信号RWに基づいてアクセス制御信号AC2、AC3を生成する。具体的には、判定信号RWが読出を指示する場合にはアクセス制御信号AC2をハイレベルに、一方、判定信号RWが書込を指示する場合にはアクセス制御信号AC3をハイレベルにする。

【0038】これにより、データDをRAM13に書き込む際（AC1がハイレベルかつRWがローレベルのとき）には、FIFOA121から書込アドレスWAと判定信号RWが出力されるとともに、FIOB122からデータDが出力される。そして、該データDはRAM13の該書込アドレスWAで指定される記憶位置に書き込まれる。一方、データDをRAM13から読み出す際（AC1がハイレベルかつRWがハイレベルのとき）にはFIFOA121から読出アドレスRAと判定信号RWが読み出されるとともに、該読出アドレスRAと該判定信号RWをRAM13に供給し、該読出アドレスRAの指定する記憶位置よりデータDを読み出して、FIFOC123に読み出されたデータDが取り込まれる。

【0039】この例にあっては、FIFOA121の段数は、1サンプリング周期内でDSP11のRAM13に対する最大アクセス回数である128あれば充分であるが、書込命令と読出命令の総数を制限すればよれより

少なくとも良い（例えば、総数96に制限して96段とする）。同様にFIO122の段数は1サンプリング周期内に実行する書込命令の最大数（128）、FIFOC123の段数は同読出命令の最大数（128）あれば良いが、それより少ない数に制限しても良い（例えば、書込48、読出80に制限し、48段、80段とする）。

【0040】ところで、CPU9は、DSP11に優先してRAM13にアクセスできるため、1サンプルクロックFsを構成する各タイムスロットの殆どをCPU9が占有することも考えられる。しかし、DSP11が正常に動作するには、一定のアクセス回数が確保されることが必要である。例えば、マイクロプログラムが図4に示すように、39回の読出命令R1～R39と17回の書込命令W1～W17とで構成されるとすれば、1サンプルクロックFs期間内にDSP11は、RAM13に56回アクセスする必要がある。この場合、128のタイムスロットのうち73以上をCPU9が占有すると、DSP11は、演算処理を実行することができない。そこで、本実施形態にあっては、送り信号発生回路124でリクエスト信号reqを生成するようにしている。このリクエスト信号reqはCPU9にウェイトをかけるものであって、リクエスト信号reqがCPU9に供給されると、CPU9は、RAM13へのアクセスを中止する。したがって、リクエスト信号reqをCPU9に供給すると、次のシステムクロックφのタイミングで制御信号CS2がローレベルとなって、この間に、DSP11はRAM13にアクセス可能となる。

【0041】リクエスト信号reqは、以下のようにして生成される。まず、各サンプル期間毎に残りのタイムスロット数が算出される。この算出処理は、送り信号発生回路124の内部に設けられた第1のダウンカウンタ（図示せず）によって行われる。第1のダウンカウンタにはサンプル周期の開始で総ステップ数（128）がロードされるようになっており、そのクロック入力端子には、システムクロックφが供給される。したがって、第1のダウンカウンタのカウント値は、各サンプル期間毎における残りのタイムスロット数を示す。

【0042】次に、各サンプル期間毎に、DSP11からRAM13へアクセスしなければならない残りのアクセス回数が算出される。この算出処理は、送り信号発生回路124の内部に設けられた第2のダウンカウンタ（図示せず）によって行われる。制御レジスタ111から1サンプルクロックFs内でRAM13にアクセスすることが必要とされるアクセス回数が供給されると、この数値がサンプル周期の開始タイミングで第2のダウンカウンタにロードされる。そして、アクセス制御信号AC1でシステムクロックφにゲートをかけ、ゲート回路の出力信号を第2のダウンカウンタのクロックとして用いる。これにより、第2のダウンカウンタのカウント値

は、DSP11からRAM13へアクセスしなければならない残りのアクセス回数を示す。

【0043】そして、第1のダウンカウンタのカウント値と第2のダウンカウンタのカウント値とを比較器(図示せず)で比較して、両者が一致すると、リクエスト信号reqを生成し、CPU9に対してウェイトをかけるようにしている。

【0044】B. 実施形態の動作

次に、図面を参照しつつ、本実施形態の動作を説明する。図6は、本実施形態に係わる電子楽器の動作の一例を示すタイミングチャートである。図6(a)は波形データのサンプル周期を示す図であり、この例では、n-1番目サンプル、n番目サンプル、n+1番目サンプルを示している。また、図6(b)は、CPU9のRAM13にアクセスするタイムスロットを示したものであり、「A」がRAM13にアクセスするスロットを、「他」がRAM13にアクセスしないスロットを示している。n-1番目サンプルにおいては、第2スロット、第4スロット、および第7スロットがRAM13にアクセスするタイムスロットである。

【0045】また、図6(c)の各サンプルの「A1~A5」は、図4のマイクロプログラムに応じて図3の波形演算部115で発生する遅延メモリ(RAM13)に対するアクセス要求の発生タイムスロットを示すものである。ここでは、第1~第3、第7スロットにおいてデータDを読み出し、第6スロットにおいてデータDを書き込む要求をそれぞれ発生している。従来のDSPではアクセス要求が発生したタイミングで遅延メモリのアクセスを行うので、これはそのまま「従来のRAMアクセス」のタイミングを示す図でもある。従来のDSPは、マイクロプログラムに従ってRAM13にアクセスする。このため、マイクロプログラムで指定される一定のタイムスロットでRAM13にアクセスする必要がある。この例においては、各サンプルの第1~第3、第6、第7タイムスロットにおいてRAM13にアクセスする必要がある。一方、CPU9のRAM13へのアクセスはランダムに生じる。したがって、CPUとDSPが同時にRAM13にアクセスする必要が生じるが、従来の構成では、DSPからのアクセスが優先され、CPUからのアクセスが制限される。例えば、第n-1番目サンプルにおいて、第2スロットと第5スロットでは、CPUとDSPが同時にRAM13にアクセスする必要が生じるので、これらのタイムスロットでは、CPUのアクセスが制限され、CPUはDSPのアクセスが行われない空き時間を使用してRAM13にアクセスする。

【0046】次に、図6(d)は、本実施形態に係わるDSP11がRAM13をアクセスする様子を示したものであり、各サンプルの「A1~A5」がRAM13へのアクセスを示すタイムスロットである。本実施形態では、上述した接続制御部12によって、CPU9のク

セスが優先されるので、DSP11は図6(b)に「他」と示したタイムスロットでRAM13にアクセスする。このため、図5(d)に示すように、DSP11は、例えば、第n番目のサンプルにおいて第0、第1、第4、第5、第7タイムスロットでRAM13にアクセスして、データDの入出力を行う。

【0047】この場合、第n番目のサンプルにおいて行われるDSP11からRAM13へのアクセスは、第n-1番目のサンプルにおいて発生したアクセス要求(図6(c)参照)によりFIFOA121に記憶された読出アドレスRA、書込アドレスWA、判定信号RWとFIFOB122に記憶されたデータDに基づいて行われる。例えば、図6(c)の第n-1サンプルの「A4」では書込要求が発生し「A1~A3」および「A5」では読出要求が発生しているため、それに対応して、図6(d)における第nサンプルの「A4」ではDSP11からRAM13への書き込みが行われ、「A1~A3」および「A5」ではDSP11によるRAM13の読み出しが行われる。第n番目のサンプルの所定のタイムスロットでDSP11が書き込みを行う場合(そのタイミングでFIFOA121から出力されるRWがローレベルのとき)、FIFOA121から該書込アドレスWAと該判定信号RWが出力されると共にFIFOB122から該データDが出力され、RAM13の該アドレスWAで示される記憶位置に該データDが書き込まれる。一方、読み出しを行う場合(RWがハイレベルのとき)には、FIFOA121から該読出アドレスと判定信号が出力され、RAM13の該アドレスRAで示される記憶位置に記憶されたデータDが読み出され、一旦、FIFOC123に格納される。そして、n+1番目のサンプルで、FIFOC123からデータDが読み出されて波形演算部115に供給される。図6(e)は、波形演算部115に供給されるデータDの様子を示したものであり、「D1~D4」がデータDの供給が行われるタイムスロットである。この例では、図6(d)におけるn番目サンプルの「A1~A3」と「A5」のアクセスでRAM13から読み出した4つのデータDがFIFOC123に格納され、n+1番目サンプルの第1、第2、第3、第7スロットで波形演算部115に供給される。これにより、波形演算部115へのデータ供給をマイクロプログラムで指定されるタイミングに調整することができる。

【0048】この実施例では、マイクロプログラムに応じてあるサンプリング周期で発生した書込要求に応じて、その次のサンプリング周期においてRAM13の指定された書込アドレスにデータDが書き込まれる。この場合、書き込まれるタイミングは1サンプリング周期遅れるが、書き込まれるアドレスは書込アドレスWAで指定されたアドレスである。一方、あるサンプリング周期で読出要求の発生時には、その次のサンプリング周期に

においてRAM 13の指定された読出アドレスからデータDが読み出され、さらにその次のサンプリング周期で読み出されたデータが波形演算部115に供給される。この場合、マイクロプログラムに応じて読出要求が発生したときにFIFO 123から波形演算部115に供給されるデータDは、2サンプリング周期前の同一タイムスロットで発生した読出アドレスRAにより読み出されたデータである。すなわち、本実施例におけるDSP 11の遅延メモリによる遅延処理では、同じディレイ長を設定した場合、通常のDSPで遅延メモリを使用してディレイを行う場合に比べて2サンプリング周期分長いディレイが得られる。従来のDSPで得られるディレイと同じ長さにするには、CPU 9が読出アドレスRAとして、操作子4等で指定されたディレイ長より2サンプリング周期分短いディレイを得るための読出アドレスRAを制御レジスタ111に設定するようにすればよい。

【0049】C. まとめ

上述したように本実施形態によれば、接続制御部12によって、CPU 9のRAM 13へのアクセスがDSP 11に優先されるので、CPU 9の処理が待たされるといったことがない。したがって、CPU 9の処理速度が低下することがなく、CPU 9の本来性能を引き出すことができる。

【0050】また、DSP 11は、CPU 9がRAM 13にアクセスしていない空きスロットを利用してRAM 13にアクセスするとともに、バッファメモリとして機能するFIFO A～Cを介してRAM 13と波形演算部115との間でデータDの入出力を行うようにしたので、RAM 13をCPU 9と共有することができ、しかもマイクロプログラムに従って波形データの演算処理を行うことができる。また、DSP 11は、空きスロットを利用してRAM 13にアクセスを行っているが、マイクロプログラムの任意のステップで発生する書込アドレスWA、データDによる書き込み、および、マイクロプログラムの任意のステップで発生する読出アドレスRAによる読み出しが可能であり、また、そこで読み出されたデータDは、該任意のステップに対応するその後のタイミングで波形演算部115に供給されるため、波形演算部115は該データDを使用しながらマイクロプログラムに従った処理を継続することができる。

【0051】また、DSP 11は、送り信号発生回路124において、各サンプル期間毎に、残りのタイムスロット数とDSP 11からRAM 13へアクセスしなければならない残りの回数とを比較し、この比較結果に基づいて、リクエスト信号reqを生成し、これにより、CPU 9にウェイトをかけるようにした。このため、CPU 9があるサンプル期間中のタイムスロットの大半を占有するような場合であっても、DSP 11はマイクロプログラムを実行するために必要なタイムスロットを確保することができる。

【0052】D. 変形例

以上、本発明に係わる実施形態を説明したが、本発明は上述した実施形態に限定されるものではなく、以下に述べる各種の変形が可能である。

①上述した実施形態において、DSP 11は、あるサンプル期間でRAM 13にアクセスするための読出アドレスや書込アドレス、RAM 13に書き込むデータD等を、FIFO A、Bに一旦格納し、次のサンプル期間でFIFO A、Bに格納されたアドレス、データD等を読み出し、これを用いてRAM 13にアクセスして読み出ししないし書き込みを行い、読み出されたデータDをFIFO 123に一旦格納し、さらにその次のサンプル期間でFIFO 123に書き込まれたデータDを読み出し、読み出されたデータDを使用して演算処理を行うようになっていた。本発明はこれに限定されるものではなく、要は、バッファメモリを使用してRAM 13へのアクセスタイミングとマイクロプログラムのタイミングを調整するものであれば、どのようなものであってもよい。例えば、1サンプル期間を0～63タイムスロット（期間X）と64～127タイムスロット（期間Y）に分割して、期間X（ないしY）でRAM 13にアクセスするための読出アドレスや書込アドレス、RAM 13に書き込むデータD等をFIFO A、Bに一旦格納し、次の期間Y（ないしX）でFIFO A、Bに格納されたアドレス、データD等を読み出し、これを用いてRAM 13にアクセスして読み出ししないし書き込みを行い、読み出されたデータDをFIFO 123に一旦格納し、さらにその次の期間X（ないしY）でFIFO 123に格納されたデータDを読み出し、読み出されたデータDを使用して演算処理を行うようにしてもよい。

【0053】この点について、図7を参照しつつ説明する。図7（a）に示すように例えば、n番目サンプルは、期間X_nと期間Y_nに分割されている。図7（b）は図6（b）と同様、CPU 9のRAM 13にアクセスするスロットを示すものである。図7（c）にマイクロプログラムに応じて発生するRAM 13に対するアクセス要求の発生タイムスロットを示す。これは、従来のDSPからRAM 13へのアクセスタイミングでもある。一方、図7（d）はこの例におけるDSP 11からRAM 13へのアクセスの様子を示す図である。この場合、DSP 11はCPU 9がRAM 13にアクセスしていない空きスロットを利用してRAM 13にアクセスしている。

【0054】この場合、期間Y_n（ないしX_n）においては、その直前の期間X_n（ないしY_{n-1}）中にFIFO A 121に格納された書込アドレスや読出アドレス、FIFO Bに格納されたデータD等を読み出して、読み出されたアドレス、データDを用いてRAM 13にアクセスし、RAM 13の書込アドレスで示される記憶領域にデータDの書き込みを行ったり、読出アドレスで

指定される記憶領域からデータDの読み出しを行い、読み出されたデータDをFIFO123に格納する。期間Y_n（ないしX_n）において読み出しが行われた場合には、その次の期間X_{n+1}（ないしY_n）において、FIFO123から順次取り出され、波形演算部115に供給される。この例によれば、各FIFOに格納するデータ量を減らすことができるので、FIFO A121, FIFO B122, およびFIFO123の段数を減らし、構成を簡易にすることができる。なお、1サンプル期間を分割する数は任意であり、CPU 9がRAM13へアクセスする頻度とDSP11がRAM13へアクセスする頻度を勘案して適宜定めればよい。この変形例では、マイクロプログラムのあるスロットで読出要求が発生したときにFIFO123から供給されるデータDは、1サンプリング周期前の同一タイムスロットで発生した読出アドレスRAにより読み出されたデータである。従って、先の実施例で述べたのと同様に、この変形例では従来のDSPに比べて1サンプリング周期分長いディレイが得られる。従来のDSPと同じ長さにしたければ、CPU9が操作子4等で指定されたディレイ長より1サンプリング周期分短いディレイを得るための読出アドレスを算出して制御レジスタ111に設定するようにプログラムしておけばよい。

【0055】②上述した実施形態においては、CPU9からのアクセスが多くDSP11によるRAM13へのアクセス回数が不足する場合には、送り信号発生回路124で生成したリクエスト信号reqを用いてCPU9にウェイトをかけてタイムスロットを確保したが、本発明はこれに限定されるものではなく、直前のサンプル期間で読み出して使用したデータDをもう一度使用するようにしてもよい。この場合には、演算の精度が多少劣化するが、CPU9が待たされるといったことは一切なくなる。なお、直前のサンプル期間（ないし直前の読出区間）に読み出したデータDをFIFO123からもう一度読み出せるようにすることは、FIFO123に1区間分の読出データDを記憶する記憶領域が確保されているので容易に実現可能である。

【0056】③上述した実施形態において、DSP11自体を波形発生器10として使用することもできる。その場合、例えば、物理モデル音源、アナログモデル音源、あるいはFM音源等のマイクロプログラムをDSP11に設定すればよい。

【0057】④上述した実施形態においては、タイムスロットの変更を行うために、FIFO A121, FIFO B122, FIFO123を使用した。本発明はこれに限定されるものではなく、バッファとして機能するメモリであればどのようなものを用いてもよい。例えば、FIFOの変わりにRAMや内部レジスタを用いてもよい。

【0058】⑤上述した実施形態において、リクエスト

信号reqの生成方法を次のように変更することもできる。1サンプリング周期内にマイクロプログラムの指示に応じて行われるDSP11のRAM13に対する全アクセス回数Xを、CPU9が予め算出する。送り信号発生回路124の内部には前述したダウンカウンタの代わりにFIFO A121に記憶されている読出アドレスと書込アドレスの総数をカウントするデータ数カウンタ（図示せず）と、そのカウント値と全アクセス回数Xを比較する比較器（図示せず）を設ける。そして、該カウント値が全アクセス回数X以上になったときにリクエスト信号reqを発生する。FIFO A121に記憶されているアドレスの総数がアクセス回数Xに達していない場合には、あるタイムスロットで発生するRAM13のアクセス要求に対応する1サンプリング周期前のスロットのアクセスは実行済みになっているはずである。さらに、別の方法として、RAM13から読み出されるFIFO123に積まれているデータDの数をカウントするデータ数カウンタ（図示せず）を設け、そのデータ数カウンタがゼロになりかつFIFO A121に1つ以上のアドレスが積まれていればリクエスト信号reqを発生するようにしてもよい。

【0059】

【発明の効果】上述したように本発明に係る発明特定事項によれば、制御プロセッサから共有メモリへのアクセスを信号処理プロセッサから共有メモリへのアクセスに優先させるように制御したので、制御プロセッサが待たされるといったことがなくなる。また、制御プロセッサと信号処理プロセッサとでメモリを共有して、構成を簡易にすることができる。

【図面の簡単な説明】

【図1】 本発明の一実施形態である電子楽器の全体構成を示すブロック図である。

【図2】 同実施形態に係わる電子楽器の主要部の構成を示すブロック図である。

【図3】 同実施形態に係わるDSPの詳細な構成を示すブロック図である。

【図4】 同実施形態に係わるマイクロプログラムの一例を示す図である。

【図5】 同実施形態に係わるアドレス発生器の回路図である。

【図6】 同実施形態に係わる電子楽器の動作を示すタイミングチャートである。

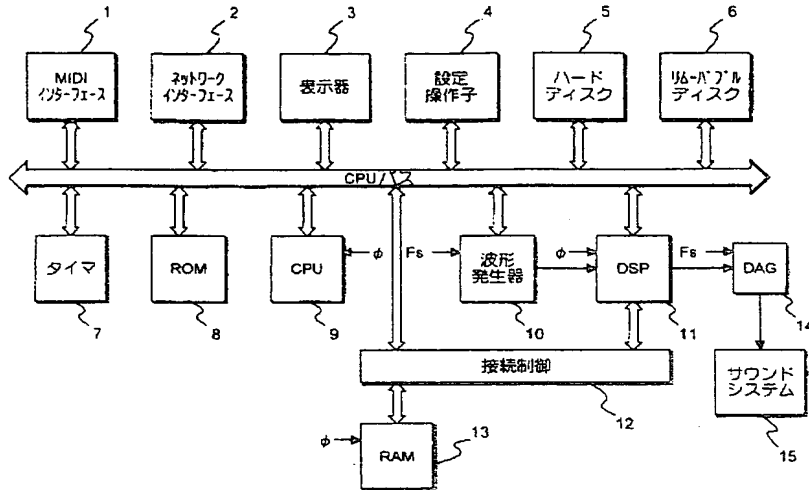
【図7】 変形例に係わる電子楽器の動作を示すタイミングチャートである。

【符号の説明】

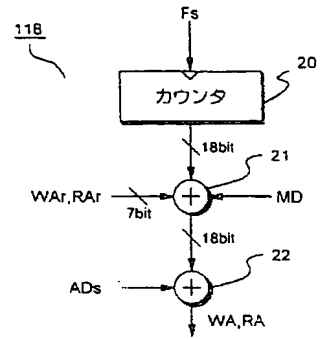
9…CPU（制御プロセッサ）、11…DSP（信号処理プロセッサ）、13…RAM（共有メモリ）、12…アクセス制御部、113…マイクロプログラムメモリ（プログラムメモリ）、114…読出回路（読出部）、115…波形演算部（演算部）、118…アドレス発生

器（アドレス発生部）、122…FIFOB（バッファ部）、123…FIFOC（バッファ部）、124…送*

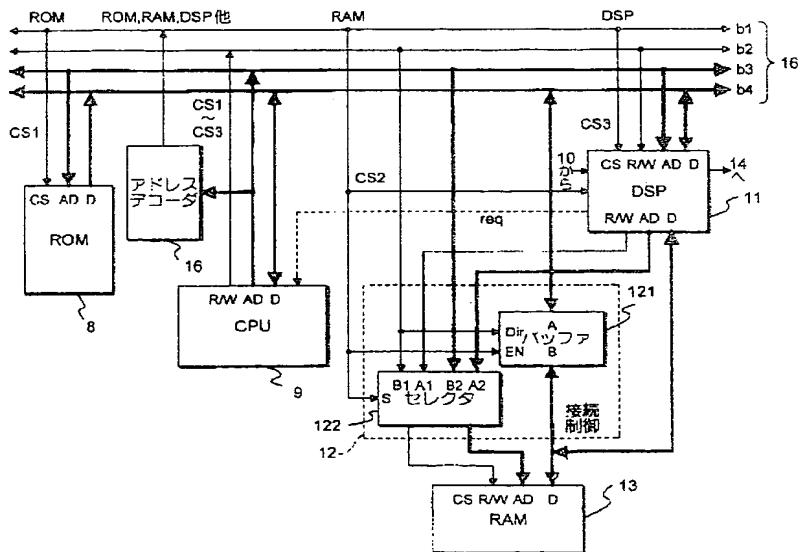
【図1】



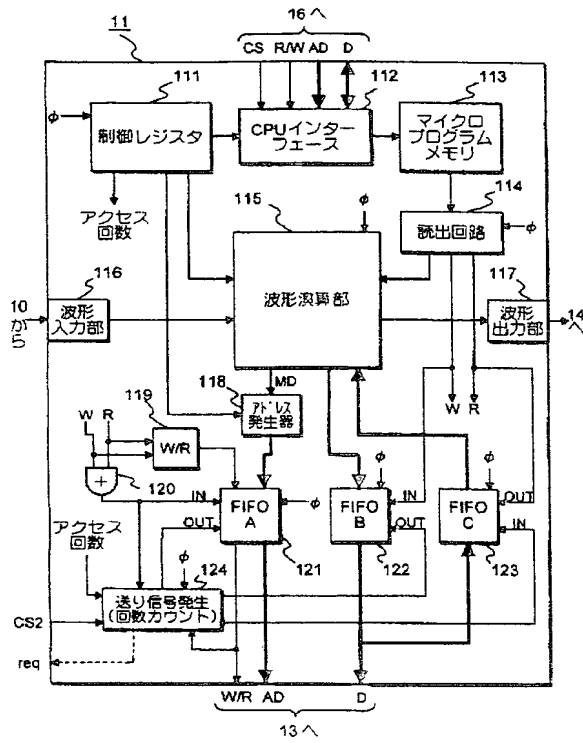
【図5】



【図2】

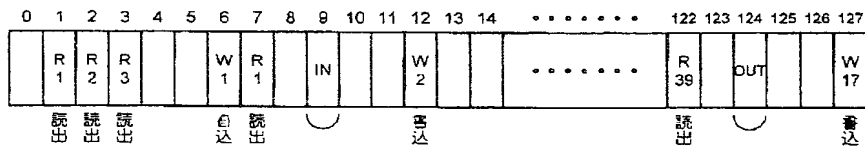


【図3】

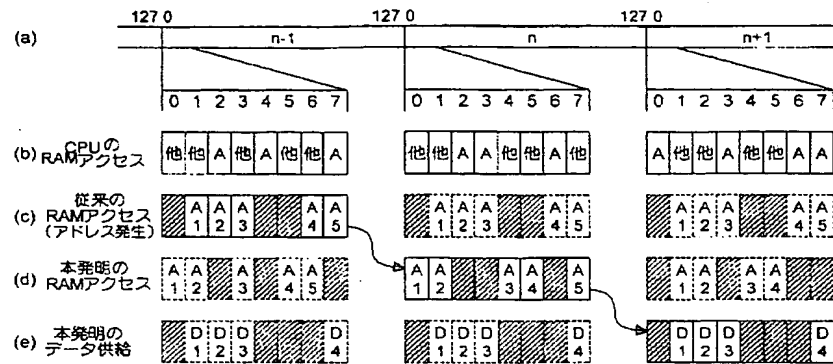


【図4】

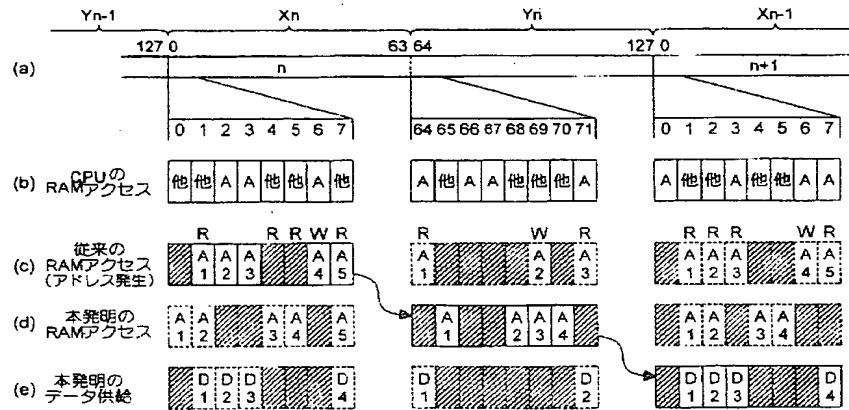
マイクロプログラム



【図6】



【図7】



THIS PAGE LEFT BLANK